**Sikuli**

Sikuli Automates anything you see on the screen using the image recognition method to identify GUI elements. Sikuli script allows users to automate GUI interaction by using screenshot

**Sikuli GUI Automation Tool**

Sikuli is a tool to automate Graphical User Interfaces (GUI) using the “Visual Image Match” method.In Sikuli, all the web elements should be taken as an image and stored inside the project / in your local macine. Sikuli will trigger GUI interactions based on the image visual match, the image which we have passed as the parameter along with all methods.

Sikuli can be very much useful to automate flash objects (which do not have ID or name). It can be useful in the situation, where we have a stable GUI (i.e. GUI components not changing).

Even Window based applications can also be automated using Sikuli. Sikuli provides very friendly Sikuli-script.jar, which can be easily used together with Selenium WebDriver. We can even automate Adobe Video/Audio player, Flash Games on the website using Sikuli. With simple API, it makes coding easier.

**Practical Uses**

* Sikuli can be used to automate Flash Objects / Flash Websites.
* It can be useful to automate the Window based application. We can automate, what we are seeing on the screen.
* It provides, simple API. i.e. all methods can be accessed using screen class objects.
* It can be easily integrated with Selenium and all other tools.
* Using Sikuli we can automate desktop applications.
* Most of the automation testing tools will not support flash-object automation (E.g. Selenium). Sikuli provides extensive support to automate flash objects.
* It uses a powerful “Visual Match” mechanism to automate desktop & flash objects.

**Benefits**

* Open-source Tool.
* One of the biggest advantages of Sikuli is that it can easily automate Flash objects.
* It makes easy to automate windows application.
* When you’re testing an application under development and you don’t know the ID/name of the elements, then you can go with Sikuli. It will check the appearance of the image and if the match found, it will interact with the image accordingly.

**Drawbacks of This Tool**

* We cannot assure you that the image match will be always accurate. Sometimes, if two or more similar images are available on the screen, Sikuli will attempt to select the wrong image.
* And if image appearance varies in pixel size, it will also result in the “Find Failed ” exception.
* Overhead of taking too many screenshots.
* If anyone of the screenshot is missing, it will affect the execution of the program.

**Screen class in Sikuli**

Screen class is the base classes for all the methods provided by Sikuli. Screen class contains predefined methods for all the commonly performed operations on screen elements such as click, double-click, providing input to a text box, hover, etc. The below is the list of commonly used methods provided by Screen class.

| **Method** | **Description** | **Syntax** |
| --- | --- | --- |
| Click | This method is used to click on an element on the screen using image name as the parameter. | Screen s = new Screen();  s.click(“QA.png”); |
| doubleClick | This method is used to double click on an element. It accepts image name as the parameter. | Screen s = new Screen();  s.doubleClick(“QA.png”); |
| Type | This method is used to provide input value to an element. It accepts the image name and text to be sent as parameters. | s.type(“QA.png”,”testdata”); |
| Hover | This method is used to hover over an element. It accepts image name as the parameter. | s.hover(“QA.png”); |
| Find | This method is used to find a specific element on the screen. It accepts image name as the parameter. |  |

## Pattern class in Sikuli

Pattern class is used to associate the image file with additional attributes to uniquely identify the element. It takes the path of the image as a parameter.

Pattern p = new Pattern(“Path of image”);

You need to add below dependency to POM.xml file

<dependency>

<groupId>com.sikulix</groupId>

<artifactId>sikulixapi</artifactId>

<version>2.0.4</version>

</dependency>